63-4-1

SWITCHING NETWORKS BY LINEAR GRAPH THEORY

by

P. Lavallee

Research Report No. PIBMRI-1146-63

for

The Air Force Office of Scientific Research
The U. S. Army Research Office
The Office of Naval Research

Grant No. AF-AFOSR-62-295

May 13, 1963

POLYTECHNIC INSTITUTE OF BROOKLYN
MICROWAVE RESEARCH INSTITUTE

ELECTRICAL ENGINEERING DEPARTMENT

Research Report No PIBMRI-1146-63

SWITCHING NETWORKS BY LINEAR GRAPH THEORY

by

P. Lavallee

Polytechnic Institute of Brooklyn
Microwave Research Institute
55 Johnson Street
Brooklyn 1, New York

Research Report PIBMRI-1146-63
Grant No. AF-AFOSR-62-295

May 13, 1963

Title Page
Acknowledgment
Abstract
Table of Contents
42 pages of text
Distribution List

P. Lavallee
Jr. Research Fellow

Approved by:

Edward Smith
Professor

for

The Air Force Office of Scientific Research
The U.S. Army Research Office
The Office of Naval Research

ACKNOWLEDGMENT

The writer is indebted to Dr. E.J. Smith for the encouragement brought about by numerous discussions with him. Also sincere thanks for Mr. W. Mow for many helpful suggestions and comments.

This work was performed while the author held a Research Fellowship during the period September 1962 through September 1963. The research was supported by the Air Force Office of Scientific Research of the Office of Aerospace Research; the Department of the Army, Army Research Office, and the Department of the Navy, Office of Naval Research under Grant No. AF-AFOSR-62-295.

ABSTRACT

This paper presents an analytic tool in the realization of circuit matrices: the method is based on forming linear trees with each row of a fundamental circuit matrix and combining them to form a tree of the graph. With this method, more difficult problems in the synthesis of contact networks by linear graph theory can be considered. A method of simultaneous synthesis is developed in which two or more switching functions are realized by sharing contacts in an optimal form. Finally the multi-terminal procedures are extended to the synthesis of non-bilateral, 1 port switching networks.

TABLE OF CONTENTS

SECTION I

Introduction

The present work discusses short-cut methods of realization of circuit matrices and applications to multi-terminal networks. The tool used is graph theory and the particular interpretation used is that of the vector spaces associated with a graph.

" The vector space V_G consists of the set of all subgraphs of the given linear graph G. The field F over which the subgraphs of G constitute a linear vector space is the field mod. 2, and addition of vectors is the ring-sum operation. The set of all subgraphs constitute a linear vector space of dimension e (where e is the number of edges). A basis of this vector space of dimension e is that defined by each edge of the graph: any subgraph can then be expressed as an e-typle(g_1, $g_2 \cdots$ g_e) of 1's and 0's. In particular the rows of the circuit matrix B are vectors of the space V_G. The set of all linear combinations of the rows of B consitute the subspace V_B: there are 2^u (including 0) vectors in V_B where u is the nullity of the graph G, and each of these is a circuit or a disjoint union of circuits of G. " (Seshu and Reed[2]).

Although the present work does not deal with interpretations and theoretical point of views, but of practical solutions, this is the point of view adopted and will be referred to in following discussion.

On examination of incompletely solved problems or problems solved by trial and error procedures arising in the graph theoretical method of solving switching functions a systematic method of solution is found which greatly reduces the number of computations required to form a solution. The Gould[1] method of realizing given switching functions has left many unsolved problems, one of which is believed to be solved in this paper and another which is a venture in the adaptation of the method to multi-terminal networks. The problem to which this technique is best adapted, is that of simultaneous synthesis of two or more switching networks.

To form a better concept of the methods developed, a very brief resume of the Gould method is presented: this resume is essentially that given in Smith, Healy and Mow[3], and so will be quoted verbation.

" Consider a topological graph G having n + 1 arcs and v vertices. Let one arc D be labelled the distinguished arc, corresponding to a source and detector. The vertices incident upon D are the terminal vertices of the graph. With each of the remaining n arcs associate a binary switching variable x_i, and let each such variable be distinct and independent. Each vertex of G corresponds to a node of the contact network. The paths through D are related to the transmission function of the network in the following way. First, obtain the collection P of paths through the distinguished arc D; let each

such path be denoted as ℓ_i. Next, form a term p_i which is the product of the elements in the path ℓ_i. Under these conditions, the sum of the product terms p_i (graph function) is in a one-to-one correspondence with the transmission function of the network. This function is called the S-C (single-contact switching function F)."

Synthesis of S-C (graph) functions

(1) The S-C function F is expressed in normal form as a sum of products in which no product term includes another product term.

(2) A loop-set matrix H is constructed from F. The columns of H correspond to the arcs of G; one column is assigned to each contact variable and to the distinguished arc **D.** The rows of H are the prime loop-set vectors of the graph which contain **D.** Let $L_{ij} = 1$ if arc J is in path i and $L_{ij} = 0$, otherwise

(3) By elementary row operations (modulo 2), the matrix H is reduced to a fundamental loop-set matrix of the type

$$B_f = \left[B_{f12} \quad I \right]$$

where I is the identity matrix and B_{f12} is a submatrix corresponding to a tree of G

(4) The normal procedure would be to form K_f orthogonal to B_f, the cut-set matrix and by elementary row operations, K_f is converted into an incidence matrix from which the graph follows immediately.

There are many methods by which step (4) is reduced: all of them are complex and are results of graphical methods as in (Okada and Young[9]), semi-graphical methods (L. Lofgren[8]), arguments based on cut-set submatrices (Mayeda[7]), arguments based on the theory of resistive n-ports (Halkias and Kim[10]). Step no (4) is the problem of realizability of a fundamental circuit matrix and an argument is derived which uses only matrix properites and two topological theorems. Hence as a really efficient substitute for the above methods, the partial linear tree method of realization is given.

SECTION II (a) Realizibility of circuit matrices

The procedures to be developed are applicable to separable or non-separable graphs. The problem of realizability has been solved by Gould[4], Guillemin[11], Lofgren[8], Mayeda[7], Okada and Young[9] and others, and more recently by Halkias and Kim[10]: the procedures given by the latter although considerably simpler than the previous ones are not extremely adaptable to mechanization on digital computer. A partial linear tree method of realization is developed.

Def. 1: A linear tree is a tree whose branches are all contained in a single path (see Fig. 1).

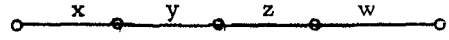


Fig. 1: Linear tree

Let B_f be a fundamental circuit matrix of order v by e and rank v: then B_f can be partitioned in a basic form

$$B_f = [U \; B_{f12}]$$

where U is a unit matrix of order v by v and B_{f12} is a matrix whose elements are members of the particular tree chosen by the diagonalization.

So far B_f has not lost its general character.

Theorem 1:

If B_{f12} contains a row (w) with non-zero elements in the columns i, j, k--- then a necessary condition for the realization of B_f, is that the matrix K formed by the columns i j k --- be realizable as a linear tree structure.

Proof: the circuit defined by a particular row (w) with non-zero elements in the columns i, j, k ... contains every element of the tree defined by edges corresponding to columns i j k: hence all branches are in a single path.

Notice that the order of the edges in the particular partial tree so defined is not known as yet: also notice that the matrix K, defined by removing from B_{f12} all columns which do not contain a "1" in the particular row we are considering is nothing else but the circuit matrix of the original B_f matrix with the tree edges not appearing in row (w) short-circuited.

Theorem 2: ordering the linear tree (due to Ash and Kim)[6]

If there exists an arrangement of the columns of the matrix K defined by row (w),

such that no two 1's are separated by one or more 0's, then this matrix is always realizable as a fundamental circuit matrix of a connected graph defined by the edges of the tree considered.

a) the tree is linear

b) the edges are ordered in the same way as the columns of the matrix defined by row (w) (after rearrangement).

The proof of this theorem goes as follows: if no two 1's are separated by one or more zeros, then the ordering of the edges contained in a row of matrix K is not contradicted by the ordering of other edges in another row of matrix K.

Example:

Consider $K = \begin{bmatrix} a & b & c \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

There exists no possible rearrangement of columns such that no two 1's are separated by one or more 0's: Hence if this matrix is a submatrix of B_f, B_f is unrealizable as a fundamental circuit matrix of a graph.

Realization procedure

Given $B_f = \bigcup B_{f12}$

a) Examine each row of B_{f12}: rows of B_{f12} which contain only two non-zero occurrences need not be examined for the reason that the linear tree formed by short circuiting all tree edges except the two edges considered, always exists (two columns always satisfy theorem 2) : consider each row containing more than two 1's in B_{f12}.

b) For each such row (w) form matrix (K) by deleting from matrix B_{f12} all columns in which 0 occurs in row (w).

c) For each such matrix K, reorder columns such that the order of the linear tree upon which matrix K is based can be found. (via Theorem 2)

d) Combine the various linear (partial) trees found in (c) and obtain the tree upon which B_f is based.

e) If reordering of a particular matrix (K) based on non (w) is impossible, the graph does not exist.

f) If combining of various partial linear trees is impossible B_f is not realizable.

This realization procedure varies markedly from that given by Gould[4], in respect that the order of the deges in a particular row of B_{f12} cound be found only after inspecting the complete matrix and forming all sets of path intersections. The

procedure resembles that of Gould's in that each partial linear tree is combined with the others.

The procedure described resembles the one given by Halkias and Kim[10] in that the various linear tree ports are formed, but the method of computation in the n-port resistive network differs completely from the method of rearrangement of columns. Finally this method is a generalization of a procedure given by Ash and Kim[6] for the class of maximal path graphs.

Several examples will illustrate the ease with which a matrix of any size can be realized, and how this method is adaptable to mechanization and more difficult problems concerning matrices with unspecified states.

In actual computation, two problems arise. The first is how do we rearrange many columns such that no two 1's are separated by one or more 0's: to do this choose any three columns, and rearrange them so as to obey the premices of theorem 2: if this is not possible, the procedure stops and no graph exists. Next take a fourth (if there is a fourth column in matrix K) and rearrange the first 3 and the fourth in the same manner, with the same conclusions, if this is not possible; we do this for all columns. The second problem is that concerned with combining each partial linear tree of B_{fl2}: it must be recognized that the ordering of columns, the order obeying theorem 2 is not unique, hence each order can give rise or not to a graph.

Notation: K(w) = matrix obtained from B_{fl2} by deleting all columns having 0's in row w.

SECTION II(b) Examples

Example 1: Determine if the following matrix given by Ash and Kim[6] is realizable, by the method described.

$$B_f = \left[\begin{array}{cccccc|cccc} e & f & g & h & i & j & a & b & c & d \\ & & & & & & 1 & 0 & 1 & 1 \\ & & & & & & 1 & 1 & 0 & 0 \\ & & & U & & & 1 & 1 & 0 & 1 \\ & & & & & & 1 & 0 & 1 & 0 \\ & & & & & & 1 & 0 & 0 & 1 \\ & & & & & & 0 & 1 & 1 & 0 \end{array}\right]$$

where U is the unit matrix

examining the rows of B_{fl2} we see that rows 1 and 3 contain more than 2 non-zero occurences: we form K_1 and K_3.

$K_1 =$

a	c	d
1	1	1
1	0	0
1	0	1
1	1	0
1	0	1
0	1	0

reordered into

d	a	c
1	1	1
0	1	0
1	1	0
0	1	1
1	1	0
0	0	1

satisfying theorem 2

: order of this linear tree is d a c

$K_3 =$

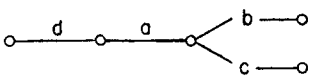
a	b	d
1	0	1
1	1	0
1	1	1
1	0	0
1	0	1
0	1	0

reordered into

b	a	d
0	1	1
1	1	0
1	1	1
0	1	0
0	1	1
1	0	0

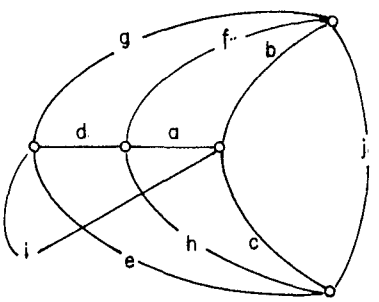
: order of this linear tree is b a d

Combining results obtained from K_1 and K_3 we get



as a tree

and



as the completed graph.

Example 2 : Determine if the following non-oriented matrix, is realizable as a fundamental circuit matrix of a graph.

	1 - - - - -	10	11	12	13	14	15
1			1	1	.	.	.
2			1	.	1	.	.
3			1	.	.	1	.
4			1	.	.	.	1
5	U		.	1	.	1	.
6			.	.	1	.	1
7			1	1	.	.	1
8			1	.	1	1	.
9			1	1	1	.	.
10			1	.	.	1	1

Note = 0's have been replaced by dots for clarity

examining the rows we see that only rows 7, 8, 9, 10 need be examined: we form K_7, K_8, K_9, K_{10}

$K_9 =$

11	12	13
1	1	.
1	.	1
1	.	.
1	.	.
.	1	.
.	.	1
1	1	.
1	.	1
1	1	1
1	.	.

reordered into

12	11	13
1	1	.
.	1	1
.	1	.
.	1	.
1	.	.
.	.	1
1	1	.
.	1	1
1	1	1
.	1	.

order of linear tree is 12, 11, 13

$K_8 =$

11	13	14
1	.	.
1	1	.
1	.	1
1	.	.
.	.	1
.	1	.
1	.	.
1	1	1
1	1	.
1	.	1

reordered into

14	11	13
.	1	.
.	1	1
1	1	.
.	1	.
1	.	.
.	.	1
.	1	.
1	1	1
.	1	1
1	1	.

order of linear tree is 14, 11, 13

$K_7 =$

12	11	15
1	1	.
.	1	.
.	1	.
.	1	1
1	.	.
.	.	1
1	1	1
.	1	.
1	1	.
.	1	1

order 12, 11, 15

$K_{10} =$

11	14	15
1	.	.
1	.	.
1	1	.
1	.	1
.	1	.
.	.	1
1	.	1
1	1	.
1	.	.
1	1	1

order 15, 11, 14

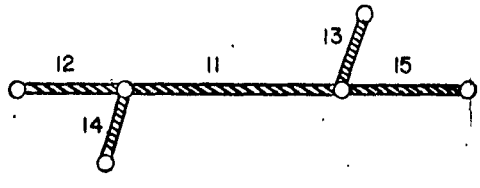
From K_7, we get 12, 11, 15

From K_8, we get 14, 11, 13

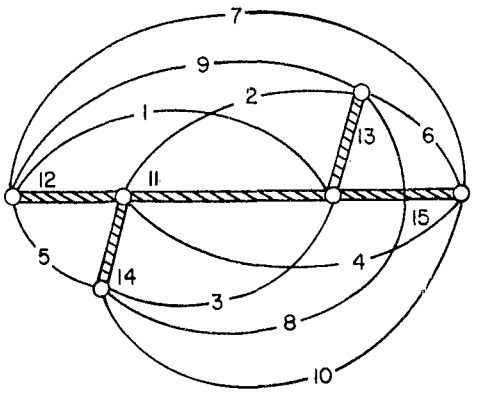
From K_9, we get 12, 11, 13

From K_{10}, we get 15, 11, 14

Combining these various partial trees we get



as the tree obtained from the partial trees



as the completed graph

Example 3: Determine if the following non-oriented matrix discussed by Gould[4] is realizable as a fundamental circuit matrix of a non-oriented graph.

$B_f =$

	1	2	3	---	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1						1	.	.	.	.	1	.	.	.	.	.	.	.	.	.
2						1	.	1	.	.	.	1	.	.	.	1	.	1	.	.
3						.	.	.	.	1	.	.	1	.	.	.	.	.	.	.
4						1	.	.	.	.	1	.	.	.	.	.	.	.	.	.
5						.	.	1	.	.	.	.	.	.	.	1	.	.	.	.
6				U		.	.	.	1	.	.	.	.	.	1	.	.	.	.	.
7						1	.	.	.	.	1	.	1	.	.	.	.	.	.	1
8						.	.	.	.	.	.	.	1	.	.	.	.	.	1	.
9						1	.	1	.	.	.	1	.	.	.	1	1	1	.	.
10						.	.	.	.	.	.	.	1	.	.	.	.	.	.	1
11						.	1	.	.	.	.	.	.	1	.	.	.	.	.	1
12						.	.	.	1	.	.	.	.	.	.	.	.	.	.	.
13						.	.	.	.	1	.	.	1	.	.	.	.	.	1	.
14						.	.	1	.	.	.	.	.	.	.	.	.	1	.	.
15						.	1	.	.	.	.	.	.	.	.	.	1	.	.	1
16						1	.	1	.	.	.	.	.	.	.	1	.	1	.	.

Examining rows of B_{f12} we see that only rows 2, 7, 11, 13, 15, 16 need be considered; we form K_2, K_7, K_9, K_{11}, K_{13}, K_{15}, K_{16}.

$K_2 =$

	17	19	23	27	29
1	1	.	.	.	.
2	1	1	1	1	1
3	.	.	.	.	.
4	1	.	.	.	.
5	.	1	.	1	.
6	.	.	.	.	.
7	1	.	.	.	.
8	.	.	.	.	.
9	1	1	1	1	1
10	.	.	.	.	.
11	.	.	.	.	.
12	.	.	.	.	.
13	.	.	.	.	.
14	.	1	.	.	1
15	.	.	.	.	.
16	1	1	.	1	1

to reorder K_2 such that no two 1's are separated by one or more 0's, take any 3 columns, reorder them; then take another column, reorder this column with the previous three already ordered and so forth, doing this we get

order 17, 29, 19, 27, 23

$K_7 =$

	17	22	24	31
1	1	1	.	.
2	1	.	.	.
3	.	.	1	.
4	1	1	.	.
5	.	.	.	.
6	.	.	.	.
7	1	1	1	1
8	.	.	1	.
9	1	.	.	.
10	.	.	1	1
11	.	.	.	1
12	.	.	.	.
13	.	.	1	.
14	.	.	.	.
15	.	.	.	1
16	1	.	.	.

stands reordered as is order 17, 22, 24, 31 or 24, 31, 17, 22 others are possible.

$K_9 =$

	17	19	23	27	28	29
1	1	.	1	.	.	.
2	1	1	.	1	.	1
3	.	.	.	.	.	.
4	1	.	.	.	.	.
5	.	1	.	1	.	.
6	.	.	.	.	.	.
7	1	.	.	.	.	.
8	.	.	.	.	.	.
9	1	1	1	1	1	1
10	.	.	.	.	.	.
11	.	.	.	.	.	.
12	.	.	.	.	.	.
13	.	.	.	.	.	.
14	.	1	.	.	.	1
15	.	.	.	.	1	.
16	1	1	.	1	.	1

notice that K_2 gave us an order of 17, 29, 19, 27, 23 all that need be done is to arrange column 28 into this ordering we get order 28, 17, 29, 19, 27, 23.

K_{11} =

	18	25	31
1	.	.	.
2	.	.	.
3	.	.	.
4	.	.	.
5	.	.	.
6	.	.	.
7	.	.	1
8	.	.	.
9	.	.	.
10	.	.	1
11	1	1	1
12	.	.	.
13	.	.	.
14	.	.	.
15	1	.	1
16	.	.	.

order 18, 31, 25

K_{13} =

21	24	30
.	.	.
.	.	.
1	.	.
.	.	.
.	.	.
.	.	.
.	1	.
.	1	1
.	.	.
.	1	.
.	.	.
.	.	.
1	1	1
.	.	.
.	.	.
.	.	.

stands ordered as is ordered 21, 24, 30

K_{15} =

18	28	31
.	.	.
.	.	.
.	.	.
.	.	.
.	.	.
.	.	.
.	.	1
.	.	.
.	1	.
.	.	.
1	.	1
.	.	1
.	.	.
.	.	.
.	1	1
.	.	.

order 18, 31, 28

K_{16} is a submatrix of K_2 hence ordering of K_2 holds for K_{16}

Gathering partial trees

From K_2, order 17, 29, 19, 27, 23

From K_7, order 24, 31, 17, 22

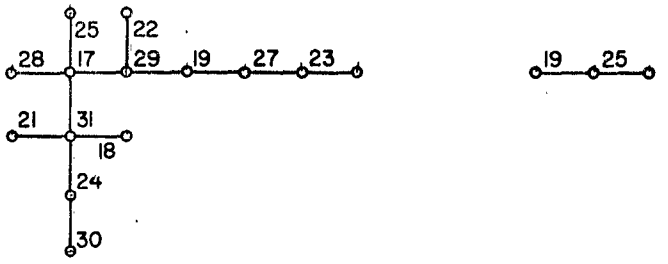
From K_9, order 28, 17, 29, 19, 27, 23

From K_{11}, order 18, 31, 25

From K_{13}, order 21, 24, 30

From K_{15}, order 18, 31, 28

Arranging each linear tree part together into the next one and doing this for each tree part we arrive at the following two tree



The completion of the graph can be done by inspection and is left as a check of the correctedness of the method: it is to be noticed that edges 19 and 25 do not appear in any K matrix: we get a two tree, the second portion having as tree the arcs 19 and 25 and thus no ambiguity.

SECTION II(c) Conclusions

As seen by the increasing complexity of the problems solved and relative ease of solution, the method of partial tree realization has distinct advantage.

As far as mechanization is concerned, we see that a simple process of extracting matrices from B_{f12}, and by successive permutations of columns, a computer could find all the successful orders of columns, i. e. all rows would contain an uninterrupted sequence of 1's: this function can be detected by a simple iterative network: as soon as a particular ordering produces an interrupted sequence, the ordering of the columns would be changed, and the process repeated.

As a final application, mention can be made of matrices arising, in connection with non-monotone realization of switching circuits: these matrices contain certain unspecified variables with constraints governing the choice (0 or 1) of these variables. This will form the next topic discussed in Section III. As one can see, the speed and accuracy of a computer could reduce even larger problems to a routine matter.

SECTION III (a) The problem of constrained matrices and non S-C functions

The full benefit of the method presented is that it forms a powerful tool in solving problem of the non S-C functions: this will be illustrated, but first a very brief discussion on the non S-C method is in order.

The method is well known and is summarized very well in Smith, Healy and Mow[3], and will be cited for completeness.

" From the given switching function, a particular set of prime implicants is selected. Assuming tentatively that the function is a S-C function a loop-set matrix B is formed from the set of prime implicants. This matrix is tested to see if the space spanned by it is acceptable; that is, linear combinations of rows which yield a loop through D, are formed and in each case, the resulting path is examined to see if it is included in the function. If all such paths are included in the function, the vector space spanned by H is acceptable and an attempt is made to realize a graph from it. If no graph is found, by the method of partial linear trees, then other normal forms of the function involving no more contacts must be tried to see if a graph results: in none of these k columns matrices yield a graph then a k + 1 column matrix is examined: this is formed by " splitting" a column corresponding to a variable x_i in such a way that the new matrix is acceptable [1].

In this process of splitting columns or of effectively duplicating the corresponding contact in the desired network, the resulting matrix may be of such a form that binary variables appear in the tree portion of the fundamental circuit matrix. We are thus faced with a multiplicity of choices (this will be demonstrated in an example) and the problem is then to find all possible graphs: as an example, in the realization the symmetric function $S_1(w, y, x, z)$ there are 9 possible 11 x 4 matrices: it was mentioned that the partial tree method could be very simply mechanized: the examinations and realization by computer is realistic, but more insight is brought into the particular problem if it is completely solved by hand.

Ex.-Give all possible realizations of $s_1(w, x, y, z)$ by graph theoretical methods. Matrix H is formed from the specified function.

$$H = \begin{array}{c} \begin{array}{cccccccccc} w & w' & x & x' & y & y' & z & z' & D & \end{array} \\ \left[\begin{array}{cccccccccc} 1 & . & . & 1 & . & 1 & . & 1 & 1 \\ . & 1 & 1 & . & . & 1 & . & 1 & 1 \\ . & 1 & . & 1 & 1 & . & . & 1 & 1 \\ . & 1 & . & 1 & . & 1 & 1 & . & 1 \end{array}\right] \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \end{array}$$

w	w'	x	x'	y	y'	z	z'	D	
1	⊡	1	⊡	1	⊡	·	1	1	1 2 3 unacceptable
1	⊡	1	⊡	·	1	1	⊡	1	1 2 4 unacceptable
1	⊡	·	1	1	⊡	1	⊡	1	1 3 4 unacceptable
·	1	1	⊡	1	⊡	1	⊡	1	2 3 4 unacceptable

(Note: □ denotes a zero by addition of an even no. of 1's)

All linear combinations of (odd number rows (modulo 2) are unacceptable: all zeroes (dots) originating from the addition of two one's (modulo 2) are enclosed by a square. To make rows 1 2 3, 1 2 4 acceptable we could split w': to make rows 1 3 4, 2 3 4 acceptable we could split y': the rules for doing this are explained in Gould[1], pages 279 - 280, and a more complete table is given in this report on page (27).

$$H_1 = \begin{array}{c} \begin{matrix} w & w_1' & w_2' & x & x' & y & y_1' & y_2' & z & z' & D \end{matrix} \\ \begin{bmatrix} 1 & \cdot & \cdot & \cdot & 1 & \cdot & b_2 & 1 & \cdot & 1 & 1 \\ \cdot & 1 & a_1 & 1 & \cdot & \cdot & b_2 & 1 & \cdot & 1 & 1 \\ \cdot & b_1 & 1 & \cdot & 1 & 1 & \cdot & \cdot & \cdot & 1 & 1 \\ \cdot & b_1 & 1 & \cdot & 1 & \cdot & 1 & a_2 & 1 & \cdot & 1 \end{bmatrix} \end{array} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix}$$

where $a_i = 0$ or $b_i = 0$ or both in H_1

By rearranging columns we get the fundamental circuit matrix

$$B_f = \begin{array}{c} \begin{matrix} w & x & y & z & w_1' & w_2' & x' & y_1' & y_2' & z & D \end{matrix} \\ \left[\begin{array}{cccc|ccccccc} 1 & & & & \cdot & \cdot & 1 & b_2 & 1 & 1 & 1 \\ & 1 & & & 1 & a_1 & \cdot & b_2 & 1 & 1 & 1 \\ & & 1 & & b_1 & 1 & 1 & \cdot & \cdot & 1 & 1 \\ & & & 1 & b_1 & 1 & 1 & 1 & a_2 & \cdot & 1 \end{array}\right] \end{array} \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix}$$

We assign to a row a particular value for the variables present in that row: for this assignment, the variables in the corresponding columns may take several values. The important point is that we extract a sub-matrix corresponding to a particular row i. e. K_w, after having assigned to the variables in that row particular states (0, 1): only this sub-matrix need be examined for realizabliity. Assign to row 1 in B_f, $b_2 = 1$:

$$K_1\ (b_2 = 1) = \begin{bmatrix} x' & y_1' & y_2' & z' & D \\ 1 & 1 & 1 & 1 & 1 \\ . & 1 & 1 & 1 & 1 \\ 1 & . & . & 1 & 1 \\ 1 & 1 & a_2 & . & 1 \end{bmatrix}$$

This is unreorderable: columns x', y_1', z' cannot be ordered as stated in theorem 2, Section II: hence we have the choice $b_2 = 1$, $a_2 = 0, 1$ unacceptable without even considering its effect on the total matrix. Since ($b_2 = 1$, $a_2 = 0,1$) can be associated with any of the three possible choices for (a_1, b_1), we have then eliminated 6 choices of binary variables

Next we try $b_2 = 0$ in row 1

$$K_1\ (b_2 = 0) = \begin{bmatrix} x' & y_2' & z' & D \\ 1 & 1 & 1 & 1 \\ . & 1 & 1 & 1 \\ 1 & . & 1 & 1 \\ 1 & a_2 & . & 1 \end{bmatrix}$$

and we examine this sub-matrix for realizability for $a_2 = 1$ and $a_2 = 0$.

Clearly, for $a_2 = 1$, in K_1, x', y_2', z' are unreorderable: hence only $a_2 = 0$, $b_2 = 0$ is a possible assignment which will satisfy sub-matrix K_1: for K, ($b_2 = a_2 = 0$) the order of the corresponding linear tree is $y_2'\ zDx'$.

Next we continue, using only successful choices in K_1 and form K_2. For $a_1 = 1$, we form K_2

$$K_2(a_1 = 1) = \begin{matrix} w_1' & w_2' & z' & D \end{matrix}$$
$$\begin{bmatrix} . & . & 1 & 1 \\ 1 & 1 & 1 & 1 \\ b_1 & 1 & 1 & 1 \\ b_1 & 1 & . & 1 \end{bmatrix}$$

reordered as $z'\ D\ w_2'\ w_1'$
or $w_1'\ z'\ D\ w_2'$

for $a_1 = 0$,

$$K_2(a_1 = 1) = \begin{array}{c} \begin{matrix} w_1' & w_2' & z' & D \end{matrix} \\ \begin{bmatrix} . & . & 1 & 1 \\ 1 & 1 & 1 & 1 \\ b_1 & 1 & 1 & 1 \\ b_1 & 1 & . & 1 \end{bmatrix} \end{array}$$

ordered as $w_1' y_2' z' D$ or $w_1' D z' y_2'$ or $w_1' z' D y_2'$ whether $b_1 = 1$ or 0

This submatrix K_2 yields no information as the nature of what the parameters a_1, b, should be, since all of them are clearly acceptable to K_2

Next we form K_3 for $b_1 = 0$

$$K_3(b_1 = 0) = \begin{array}{c} \begin{matrix} w_2' & x' & z' & D \end{matrix} \\ \begin{bmatrix} . & 1 & 1 & 1 \\ a_1 & . & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & . & 1 \end{bmatrix} \end{array}$$

for $a_1 = 1$, columns $w_2' x' z'$ are un-reorderable
for $a_1 = 0$, we get order $zDx' w_2'$: hence for $a_1 = 1$, $b_1 = 0$, no realization possible.

All that is left to be determined now is b_1

for $b_1 = 1$, form K_3

$$K_3(b_1 = 1) = \begin{array}{c} \begin{matrix} w_1' & w_2' & x' & z' & D \end{matrix} \\ \begin{bmatrix} . & . & 1 & 1 & 1 \\ 1 & a_1 & . & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & . & 1 \end{bmatrix} \end{array}$$

and columns $w_1' x' z'$ are un-reorderable hence no matter what a_1 is for $b_1 = 1$, we do not get a realization. Hence only $a_1 = 0$, $b_1 = 0$ is a consistant assignment. So final successful choices of binary variables are:

$a_2 = 0, \quad b_2 = 0$
$a_1 = 0, \quad b_1 = 0$

and these are unique

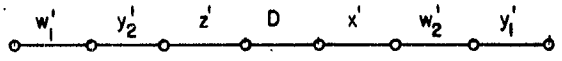
From K_1 we get order: $y_2' z' D x'$

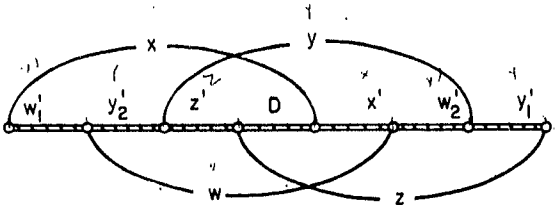
From K_2 we get order: $w_1' D z' y_2'$ or $w_1' y_2' z' D$

From K_3 we get order: $z' D x' w_2'$

From K_4 we get order: $z' D x' w_2' y_1'$

Combining these linear trees

we get  as a tree

and  as the graph.

Notice that K_4 was not even used: we eliminated all possible choices of binary variables before using up all our analytic tools. It remained then only to check if successful binary choices made K_4 contradictory or not. In this case it did not. It is thus demonstrated that the method is applicable, but is generally tedious: the high speed of a computer would in this case be well adapted to such problems. The advantage of the method is that only submatrices need be examined whereas where no method existed, it was necessary to examine the complete matrix for each particular combination of binary variables: a summary of a synthesis procedure of this type now can be given.

SECTION III(b) Realization procedure

For the case of fundamental circuit matrices with constrained binary variables; to answer the problem of finding all possible realizations, we give this procedure:

1) Extract a K_w matrix (corresponding to row w) having beforehand assigned a particular value to variables in this row: examine the realizability of this matrix with this assignment.

2) Give another assignment ot the binary variable in that some row and proceed as in 1.

3) Do this for all rows, but allowing only values of binary variables that were successful in step 1 and 2, to appear in step 3.

4) Corresponding to each successful set of binary variables, combine all linear trees: for each set there may or may not exist a graph, as discussed in the synthesis by partial linear trees.

SECTION IV (a) Multi-terminal networks

As a mechanized way of finding if a matrix yields a graph has been demonstrated, this leads to consideration of problems of higher order of complexity such as multi-terminal contact networks. There are different methods yielding multi-terminal networks: we can picture this type of problem as a multi-input, multi-output block box problem. It has been stated by authors that methods of graph theory are applicable to this problem, and a resume of the work that has been proposed shall be given. The primary reason for this investigation is that there eixsts no method of yielding minimal multi-terminal switching networks: the graph theoretical methods do yield in most instances minimal solutions (in terms of numbers of contacts) for a single input-output system: i. e. a 1 port. The question is then: does the graph method yield a so-called minimal solution for multi-terminal system.

Gould and Lofgren both proposed methods that shall be described briefly. The technique discussed by Gould[1] is that for each of n outputs realizing n switching functions F_n, there exists a detector D_n and all of these detectors have in common a source D_o: the vectors or loop-sets in F_i should have 1-entries for D_o and D_i respectively; furthermore, it should be verified that every vector in the space which has 1-entries for D_i and D_o but not for D_j, either is a block-loop set or comes a prime inplicant vector of F_i. Hence every vector in the space which has a D_o 1-entry will also have either a 1-entry for D_i or D_j but not for both. Now it is stated that this guarantees that the source and detector share a common terminal: that this is not so is proved simply by the argument all that the D_i's and D_o guarantee is that they form closed paths each path containing D_o, but with no reference whatsoever to the fact that the D_i's and D_o must form a star point: if this method did specify a star point for D_o and all the D_i's, the problem would be simplified. Once the matrix has been found acceptable, then the realization problem from thereon in, follows the partial linear tree method.

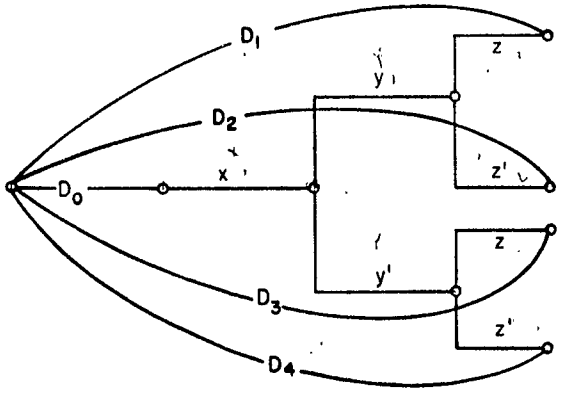
The second contributor Lofgren[8] recognizes that fact that the source and detectors need not form a star point: it is therefore purely accidental that they should form a star point, if no further restriction on the detectors, are imposed.

We now differenciate between a common terminal graph and a non-common terminal graph: the specification of a common terminal (all detectors originating from the same node) will be shown to be a special case of the more general situation. From the black box approach this is easily seen: we have a set of input and a set of detectors at the output: no matter where the detectors originate in the box, we known that a certain combination of inputs, corresponding to a vector in F_i, occurs if D_i is energized.

Consider for example the specification

$$T_{10} = xyz, \quad T_{20} = xyz', \quad T_{30} = xy'z', \quad T_{40} = xy'z$$

Realizing this specification by the disjunctive tree method we get the following common terminal graph.



We now try to solve this probelm by the methods of graph theory: first we change the specification to

$$T_1 = xyz, \quad T_2 = xyz', \quad T_3 = xy'z', \quad T_4 = xy'z$$

and search for a solution such that as far the the detectors are concerned we get the same switching functions. We set up the H matrix and check for acceptability of H. If a path has one or more detectors in it (and the source D_o), it must cover a vector in each function specified by the detectors in that path or else represent a block-loop set.

	x	y	y'	z	z'	D_1	D_2	D_3	D_4	D_o		
H =	1	1	.	1	.	1	.	.	.	1	1	
	1	1	.	.	1	.	1	.	.	1	2	
	1	.	1	.	1	.	.	1	.	1	3	
	1	.	1	1	.	.	.	.	1	1	4	
	1	⊡	1	1	⊡	1	1	1	.	1	1 2 3	unacceptable
	1	⊡	1	⊡	1	1	1	.	1	1	1 2 4	"
	1	1	⊡	1	⊡	.	1	1	1	1	2 3 4	"
	1	1	⊡	⊡	1	1	.	1	1	1	1 3 4	"

Sum rows	Contacts to be split
123	y, z'
124	y, z
234	y', z'
134	y', z

We can thus satisfy sum rows 123, 234 by splitting z' and sum rows 124, 134 by splitting z. An alternate arrangement is splitting y and y': the former shall be used; the splitting is of the form $\begin{bmatrix} 1 & a \\ b & 1 \end{bmatrix}$, a = 0, or b = 0, or both

Since the matrix is in diagonalized form, we thus have reduced the problem to that of realizing a fundamental circuit matrix with constrained variables.

$$B_f = \begin{array}{c} \begin{matrix} D_o & x & y & y' & z_1 & z_2 & z'_1 & z'_2 & D_1 & D_2 & D_3 & D_4 \end{matrix} \\ \begin{bmatrix} 1 & 1 & 1 & . & 1 & a_1 & . & . & 1 & . & . & . \\ 1 & 1 & 1 & . & . & . & 1 & a_2 & . & 1 & . & . \\ 1 & 1 & . & 1 & . & . & b_2 & 1 & . & . & 1 & . \\ 1 & 1 & . & 1 & b_1 & 1 & . & . & . & . & . & . \end{bmatrix} \end{array}$$

Where the a's = 0 or b's = 0 or both

There exists nine possible matrices and the problem is to find for which assignment is B_f realizable. Following the procedures developed in section II we find:

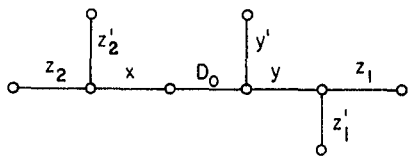
For the case $a_1 = a_2 = 1$, $b_1 = b_2 = 0$

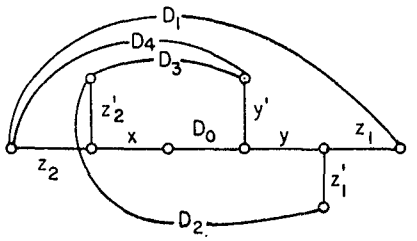
we get from K_1 order $z_2 \times D_o y z_1$

from K_2, order $z'_2 \times D_o y z'_1$

from K_3, order $z'_2 \times D_o y'$

from K_4, order $z_2 \times D_o y'$

and  as a tree

and  as the complete graph.

Sum rows	Contacts to be split
123	y, z'
124	y, z
234	y', z'
134	y', z

We can thus satisfy sum rows 123, 234 by splitting z' and sum rows 124, 134 by splitting z. An alternate arrangement is splitting y and y': the former shall be used: the splitting is of the form $\begin{bmatrix} 1 & a \\ b & 1 \end{bmatrix}$, a = 0, or b = 0, or both

Since the matrix is in diagonalized form, we thus have reduced the problem to that of realizing a fundamental circuit matrix with constrained variables.

$$B_f = \begin{array}{c} \begin{matrix} D_o & x & y & y' & z_1 & z_2 & z'_1 & z'_2 & D_1 & D_2 & D_3 & D_4 \end{matrix} \\ \begin{bmatrix} 1 & 1 & 1 & . & 1 & a_1 & . & . & 1 & . & . & . \\ 1 & 1 & 1 & . & . & . & 1 & a_2 & . & 1 & . & . \\ 1 & 1 & . & 1 & . & . & b_2 & 1 & . & . & 1 & . \\ 1 & 1 & . & 1 & b_1 & 1 & . & . & . & . & . & 1 \end{bmatrix} \end{array}$$

Where the a's= 0 or b's = 0 or both

There exists nine possible matrices and the problem is to find for which assignment is B_f realizable. Following the procedures developed in section II we find:

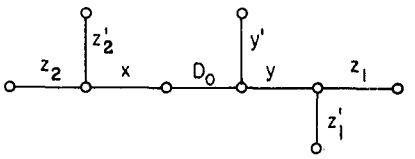
For the case a_1 $a_2 = 1$, $b_1 = b_2 = 0$

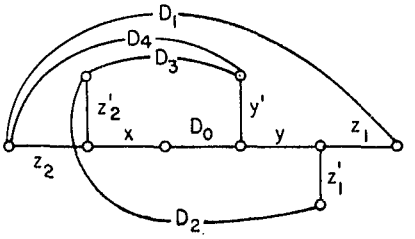
we get from K_1 order $z_2 \times D_o y z_1$

from K_2, order $z'_2 \times D_o y z'_1$

from K_3, order $z'_2 \times D_o y'$

from K_4, order $z_2 \times D_o y'$

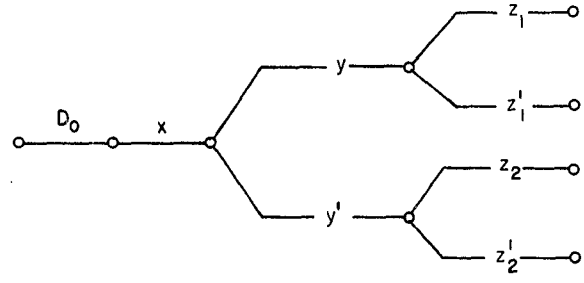
and  as a tree

and  as the complete graph.

This was for a particular choice of binary variables, but it illustrates the point that as far as the output is concerned, it behaves in the same manner as the common point detector type of graph. It is to be noticed, that as a special case of binary variables $a_1 = a_2 = b_2 = b_1 = 0$, we get from the matrix H, the following trees.

from row 1, K_1 : order : $D_o x y z_1$
from row 2, K_2 : order : $D_o x y z'_1$
from row 3, K_3 : order : $D_o x y' z'_2$
from row 4, K_4 : order : $D_o x y' z_2$

and



as a tree

This is exactly the tree obtained by the disjunctive tree method. Thus it has been demonstrated that the common terminal is a special case arising in the synthesis by graph theory of multi-terminal switching networks.

SECTION IV (b) Simultaneous synthesis

The conclusions to be drawn on this seemingly pointless realization are two-fold: first the method has demonstrated that a star point of detectors specification is a specialized specification and secondly that a method of simultaneous synthesis of switching functions can now be evolved.

The usual problem dealt with in multi-terminal switching networks is that of the common terminal detector type. A number of methods are available: among others there is the assumed form method, the boolean matrix method, the disjunctive tree method; all are well known (Caldwell[12]) and each best adapted to specific types of problems.

The problem of interest and which can be solved is the following: given any two or more systems, could it not be possible to utilize some of the components of systems ($A_i \ldots A_s$) and minimize systems ($B_i \ldots B_j$) and vice-versa. The systems here would represent the realization of particular switching functions. Given two or more switching functions, is it more economical to realize each function separately or simultaneously.

A systematic procedure is developed: the price paid for absolute minimality and generality is usually an abundance of additional complications: the price paid in this method is that the physical size of the problem increases beyond proportion for moderate size problems. Nevertheless, each step shall be discussed and systematized.

Step 1: Multiple output prime implicants

Since the single output method dealt with the function reduced to one of its prime implicant form (if more than one exists), the same technique but with the generalization to multiple output prime implicants is presented. The treatment of the generalized case is essentially that of Vandling[13], and will briefly be summarized. Given q different single-output functions of n variables, each is expressed in sum of product form as

$$F_i = \sum_{j=0}^{2^n-1} Z_{i,j}\, f_j \qquad Z_{ij} = 1 \text{ or } 0,\ i = 1, 2 \ldots q$$

:where $f_j(x_1 - - x_n)$ is defined by the arithmetic sum

$$j = b_1 + 2b_2 + 4b_3 \ldots 2^{n-1} b_n$$

and $b_p = 1$ if x_p appears in f_j, $b_p = 0$ if x'_p appears in f_j

:where Z_{ij}'s are assigned values of 1 or 0 depending on whether or not F_i is to have a 1 or 0 output for input values described by f_j. If we associate to each single-output function an auxiliary variable D_i, we can express the multiple output switching function F as

$$F = \sum_{i=1}^{q} \sum_{j=0}^{2^n-1} D_i Z_{ij} f_j \qquad \text{where } Z_{ij} = 1 \text{ if } f_j \text{ is in } F_i$$

and 0 otherwise: by definition each $D_i f_j$ is a multiple output minterm.

Ex.: $F_1 = x_1x_2 + x'_1x'_2$

$F_2 = x_1x_2 + x'_1x'_2$

$F_3 = x'_1x_2 + x'_2x_1$

then $F = D_1x_1x_2 + D_1x_1'x_2' + D_2x_1x_2 + D_2x_1'x_2 + D_3x_1'x_2 + D_3x_2'x_1$

To obtain the prime inplicants, a set of 2^{q-1} function G_m are defined, each of which corresponds to a subset of the set of all sum of products appearing in the q function F_i.

$$G_1 = F_1$$
$$G_2 = F_2$$
$$G_3 = F_1 \cap F_2$$
$$G_4 = F_3$$
$$G_5 = F_1 \cap F_3$$
$$G_6 = F_2 \cap F_3$$
$$G_7 = F_1 \cap F_2 \cap F_3$$
$$\vdots$$
$$G_{2^{q-1}} = F_1 \cap F_2 \cap F_3 \cdots \cap F_q$$

where $\cap$ means the usual intersection symbol.

where the subscripts m are assigned according to:

$$m = a_1 + 2a_2 + 4a_3 \ldots\ 2^{i-1}a_i \ldots\ a^{q-1}a_q$$

and $a_i = 1$ if G_m is a function of F_i

ex.: for $m = 15 = 1 + 2 + 4 + 8$ $\quad G_{15} = F_1\ F_2\ F_3\ F_4$

For each single-output prime implicant S_q of the set G_m there is a correponding multiple output prime implicant (M) of F such that

$$M = (h_1D_1 + h_2D_2 \text{ --- } h_qD_q)\ S(G_m)$$

where $h_i = 1$ if G_m is a function of F_i and 0 otherwise.

The next step in the determination of the multiple prime implicants is to set up a prime implicant chart, the rows of which correspond to M, the columns of which correspond to a minterms in specific F_i's and we proceed as in the ordinary prime implicant chart with an additional relation: a multiple output prime implicant M, implies a single output minterm B if

a) every literal in M appears in B, no variable appearing primed and unprimed

b) every auxiliary variable in B appears in M

The reason for this discussion will become apparent after an example.

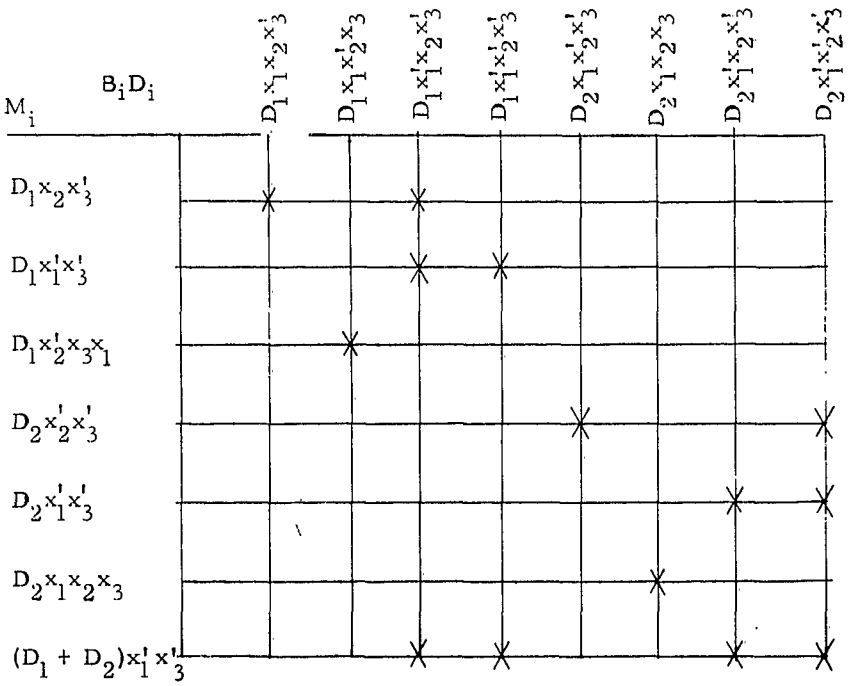
Ex. :

$$G_1 = x_1x_2x_3' + x_1x_2'x_3 + x_1'x_2x_3' + x_1'x_2'x_3' \quad F_1$$

$$G_2 = x_1x_2'x_3' + x_1x_2x_3 + x_1'x_2x_3' + x_1'x_2'x_3' = F_2$$

$$G_1 G_2 = G_3 : x_1'x_2x_3' + x_1'x_2'x_3'$$

Prime implicants S	Prime implicants M
F_1 : x_2x_3'	$D_1x_2x_3'$
$x_1'x_3'$	$D_1x_1'x_3'$
$x_1x_2'x_3$	$D_1x_2'x_3x_1$
F_2 : $x_2'x_3'$	$D_2x_2'x_3'$
$x_1'x_3'$	$D_2x_1'x_3'$
$x_1x_2x_3$	$D_2x_1x_2x_3$
F_3 : $x_1'x_3'$	$(D_1 + D_2)x_1'x_3'$



and from the table we find that

$$F = D_1x'_2x_3x_1 + D_1x_2x_3' + (D_1 + D_2)x'_1x'_3 + D_2x'_2x'_3 + D_2x_1x_2x_3$$

Now in the graph theory approach a vector $(D_1 + D_2)x'_1x'_3$ would appear in the matrix as

$$\begin{bmatrix} x'_1 & x'_3 & D_1 & \cdot & D_o \\ x'_1 & x'_3 & \cdot & D_2 & D_o \end{bmatrix}$$

and the modulo 2 sum of these vectors would yield a vector D_1D_2: although at first sight we could not even check such a combination (sum row) since its D_o entry is zero, we get a contradiction in specification: D_1D_2 implies that $D_1 = D_2$ since the detectors would be in parallel. The prime implicant chart does give us an indication of what contacts to split in order to preserve the proper specification. This then indicates that not only linear combinations (nod. 2) of 3, 5, 7 ... rows have to be checked, but also those rows which have identical primary variables in them, but different D_i's. In this example we thus know beforehand that either x'_1 or x'_3 must be split.

Generalizing if D_i, D_T and D_K share a path set, in $\underline{F}$, then we know that a contact must be split in three.

Step 2 Optimum contact splitting

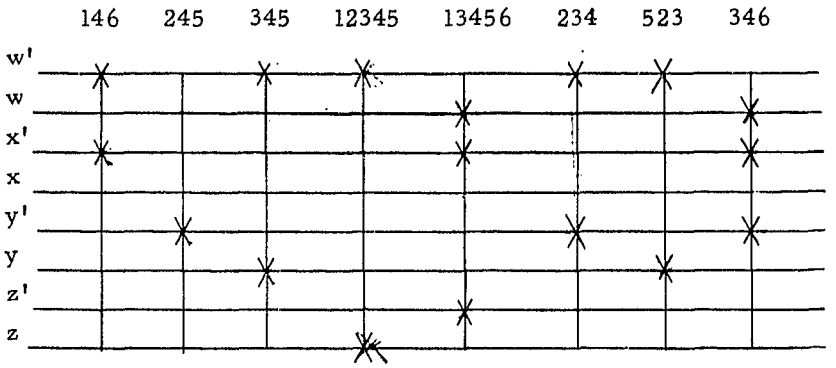
Once a minimal form for the multiple output function F has been found, to each vector we assign an auxiliary variable D_o and each vector thus affixed with D_o (source). becomes a row of a matrix H. : if $F = (D_1 + D_2)\, f_i$

$$\text{then} \qquad H = \begin{bmatrix} f_i & D_1 & D_o & \cdot \\ f_i & \cdot & D_o & D_2 \end{bmatrix}$$

A problem that arises in testing H, is that of splitting elements in such a way that a minimum number of elements are needed. This is dealt in a method analogous to the prime implicant chart: for example consider the following table arising in a simultaneous synthesis of two switching functions T_1, T_2 each having three vectors.

row sum	contact split to make row sum legitimate
146	w', x'
245	y'
345	w', y
346	w, x', y'
12345	w', z
13456	w, x', z'
423	w', y'
523	w', y

Consider the following construction



Reasoning as in the prime implicant chart, we conclude that y' is an essential element to be split since it alone covers column 245. y' covers 245, 234, 346, hence these columns can be ruled out. No matter what other row is chosen it cannot cover all other columns, hence we conclude that a minimum of 3 but no more rows are needed to cover all columns: there are many possible combinations: all of these must include y' as an example we have y', w', z'

A table which is an enlarged version of that given by Gould[1] is presented, giving rules of duplication of contacts.

Duplication rules

Row sum specified	General configuration		Restriction on variables
1. $\xi + \eta$	ξ η	1 a b 1	a = 0 or b = 0 or both
2. $\xi + \eta$ $\xi + \zeta$	ξ η ζ	1 a c b d 1	if a = 0 b = 1 if a = 1, b = c', d = 0
3. $\xi + \eta$ $\xi + \zeta$ $\eta + \zeta$	ξ η ζ	1 a b' a' b 1	a = 0 or b = 0 or both
4. $\xi + \eta$ $\xi + \zeta$ $\xi + \eta + \zeta$	ξ η ζ	1 a b 1 b 1	a = 0 or b = 0 or both
5. $\xi + \eta$	ξ η ζ y	1 a b 1 e f g h	a = 0 or b = 0 or both e = 1 or f = 1 or both g = 1 or h = 1 or both
6. $\xi + \eta$ $\xi + \zeta$	ξ η ζ y	1 a c b d 1 e f	same as 2 with e = 1 or f = 1 or both
7. $\xi + \eta$ $\xi + \zeta$ $\eta + \zeta$	ξ η ζ y	1 a b' a' b 1 e f	same as 3 with e = 1 or f = 1 or both
8. $\xi + \eta$ $\xi + \zeta$ $\eta + \zeta$ $\xi + \eta + y + \zeta$	ξ η ζ y	1 a b' a' b 1 e f	same as 3 with e = 1 or f = 1 or both
9. $\xi + \eta$ $\xi + \zeta$ $\xi + \eta + \zeta + y$	ξ η ζ y	1 a c b d 1 e f	same as 2 with if a=0, and c=d, e=0 or f=1 if a=0, and c≠d, e=1 or f=1 if a=1, and c=0, e=0 or f=0 if a=1, and c=1, e=1 or f=1
10. $\xi + \eta$ $\xi + \eta + \zeta + y$	ξ η ζ y	1 a b 1 e f g h	same as 1 with if a=0, b=1, f=h, or g≠e if b=0, a=1, f≠h, or g = e if a = b = 0, f=h, or g=e
11. $\xi + \eta$ $\xi + \zeta$ $\xi + \zeta + \eta$	ξ η ζ y	1 a b 1 b 1 e f	a = 0 or b = 0 or both e = 1 or f = 1 or both
12. $\xi + \eta$ $\xi + \zeta$ $\xi + \eta + \zeta$ $\xi + \eta + \zeta + y$	ξ η ζ y	1 a b 1 b 1 e f	a = 0 or b = 0 or both if a = 0, e = 0 or f = 0 if a = 0, e = 0 or f = 1

SECTION IV (c) Formal synthesis procedure

1- From the specification $\{T_i\}_i$ we form the prime implicants for each T_i and then set up a prime implicant chart for the multiple-output function F.

2- From an examination of F, we determine the absolute minimum numbers of contacts to be split in order that $T_i \neq T_j$.

3- Form matrix H, from each vector in F and check for acceptability: all unacceptable paths and all sums of paths which were shared in F are written below H.

4- Split contacts (if needed) according to the ordinary acceptability requirements, and requirements of step 2: use is made of the duplicating chart and the table of duplication rules.

5- The synthesis procedure follows from here, the same steps as in the single output synthesis as discussed in Section II.

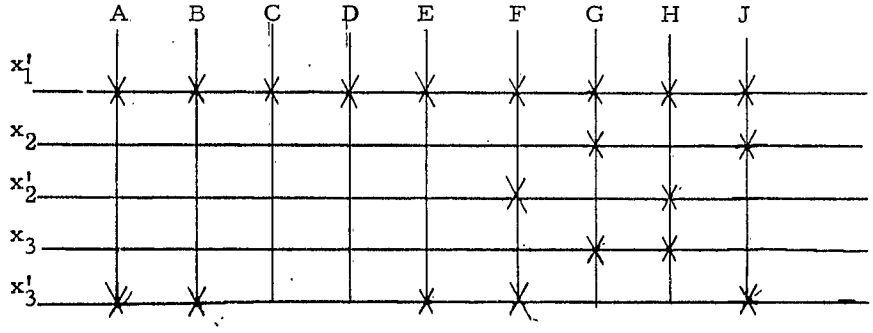
SECTION IV (d) An example

Synthesize the function $F = D_1x_2'x_3x_1 + D_1x_2x_3' + (D_1+D_2)x_1'x_3' + D_2x_1x_2x_3 + D_2x_2'x_3'$

We assume that the specification $\{T_1, T_2\}$ has been transformed in F by the methods discussed previously. Next H is formed from F.

	x_1	x_1'	x_2	x_2'	x_3	x_3'	D_1	D_2	D_o
1	1	.	.	1	1	.	1	.	1
2	.	.	1	.	.	1	1	.	1
3	.	1	.	.	.	1	1	.	1
4	.	1	.	.	.	1	.	1	1
5	.	.	.	1	.	1	.	1	1
6	1	.	.	.	1	.	.	1	1
34	.	⊡	.	.	.	⊡	1	1	.
134	1	⊡	.	1	1	⊡	.	1	1
234	.	⊡	1	.	.	1	.	1	1
345	.	⊡	.	1	.	1	1	.	1
346	1	⊡	1	.	1	⊡	1	.	1
12345	1	⊡	1	⊡	1	⊡	1	.	1
12346	⊡	⊡	⊡	1	⊡	1	1	.	1
13456	⊡	⊡	1	⊡	⊡	1	.	1	1
23456	1	⊡	⊡	1	1	⊡	.	1	1

Sums rows		Contacts to be split
A	34	x_1' or x_3'
B	134	x_1' or x_3'
C	234	x_1'
D	345	x_1'
E	346	x_1' or x_3'
F	12345	x_1' or x_2' or x_3'
G	12346	x_3, x_2, x_1'
H	13456	x_3, x_2', x_1'
J	23456	x_1', x_2, x_3'



We try a solution splitting x_1' since it covers all sum rows.

	x_1	x_{11}'	x_{12}'	x_2	x_2'	x_3	x_3'	D_1	D_2	D_o
1	1	.	.	.	1	1	.	1	.	1
2	.	.	.	1	.	.	1	1	.	1
3	.	1	a	.	.	.	1	1	.	1
4	.	b	1	.	.	.	1	.	1	1
5	.	.	.	.	1	.	1	.	1	1
6	1	.	.	1	.	1	.	.	1	1

a = 0 or b = 0 or both

From the specification F we know that at least one contact must be split (either x'_1 or x'_3 must be duplicated). From H, we eliminate row 2 being linearly dependent of row sum 156. There exists three possible solutions of the matrix H, due to the constrained variables. Each choice fails. The method is so built that it searches for absolute minimality: next a solution by duplicating two contacts is tried. From the duplicating chart we must use x'_1 since it covers columns C and D by itself, and in conjunction with x'_1 we duplicate x_2.

Solution for x'_1 and x_2 fail: similarly solutions with x'_1 and x'_2, x'_1 and x_3, x'_1 and x'_3 all fail.

A solution by duplicating three contacts is is tried: using x'_1 and x'_2 and x_2 we get an immediate solution: the splitting is of the type $\begin{bmatrix} a & 1 \\ 1 & b \end{bmatrix}$. Diagonlization in this case is very simple and we end up with

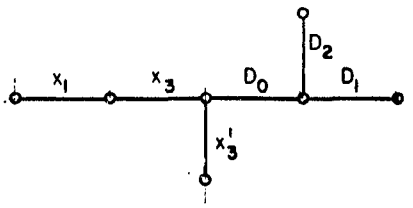
$$B_F = \left[\begin{array}{cccccc|cccccc} x'_{11} & x'_{12} & x_{21} & x_{22} & x'_{21} & x'_{22} & x_1 & x_3 & x'_3 & D_1 & D_2 & D_o \\ \hline . & . & . & . & 1 & . & 1 & 1 & . & 1 & . & 1 \\ . & . & 1 & . & . & . & . & . & 1 & 1 & . & 1 \\ 1 & . & . & . & . & . & . & . & 1 & 1 & . & 1 \\ . & 1 & . & . & . & . & . & . & 1 & . & 1 & 1 \\ . & . & . & . & . & 1 & . & . & 1 & . & 1 & 1 \\ . & . & . & 1 & . & . & 1 & 1 & . & . & 1 & 1 \end{array}\right]$$

From K_1, order $x_1 x_3 D_o D_1$

" K_2 " $x'_3 D_o D_1$

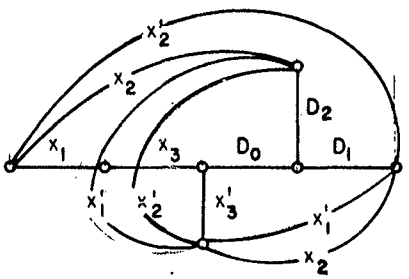
" K_4 " $x'_3 D_o D_2$

" K_5 " $x_1 x_3 D_o D_2$



as the tree

and



as the find graph F

As a conclusion to this example, it can be added that for a quick solution having a 25 % or more savings in contacts we could at the outset of examining the contact splitting chart, choose more than the minimum number and try for a solution. The more contacts are split, the easier the solution becomes. It can be verified that the separate synthesis takes 12 contacts and the simultaneous solution presented, takes 9.

SECTION V (a) Contact diodes networks

The main disadvantages of the method of simultaneous synthesis discussed in Section IV, is the physical size of the problem, and the search for absolute minimality. This search tends to be of a repetitive nature and rather than present more examples which are theoretically and practically capable of being solved by repetitions of the processes discussed previously, a very special set shall be investigated.

Suppose the specification of the system is of the type

$$\{T_{12},\ T_{21}\}$$

In other words we shall be dealing with non-bilateral one-port switching functions, and all the methods discussed will apply. Previous work on this subject, by graph methods, has been done by "Smith, Healy and Mow"[3]. The problem is solved for a special set of functions, realizable by maximum loop techniques. The synthesis depends on the final orientation of a modulo 2 fundamental circuit matrix B_f, by setting all non-zero entries of the matrix to +1, provided the matrix in maximum loop form.

$B_f = [I\ B_{f12}]$ is in maximum loop form if B_{f12} contains a row which has non-zero entries for each column of B_{f12}.

The solution adopted shall use only the field of integers modulo 2 and some additional methods of making the H matrix acceptable.

Applying previous methods, it is seen that two functions are specified, and thus two detectors are needed. An additional constraint is added in the specification in that both detectors are connected to the same nodes: this implies the use of oriented or non-bilateral detectors, oriented in the same direction with respect to the path they form. Since we are dealing with a one port, no source shall be provided. The synthesis is started by the usual single-contact and single diode technique. The method of converting $\{T_{12},\ T_{21}\}$ into F (multiple output function) using a literal r to designate the diode is discussed in Smith, Healy and Mow[3]: examples will illustrate the method.

SECTION V (b) Simplification and constraint

A simplification due to the unilateral element is provided in the testing procedures for the acceptability of the loop-set matrix H. In checking if the original loop-set matrix is acceptable, we may come up with a path which is unacceptable (does not cover a row $_j$ of H, or does not represent a blocked loop-set). For example, take a path $xyrD_1$. To make this path acceptable, we can choose the orientation of D_1 and r to be opposite in this particular path. This would represent a blocked loop-set: this also implies that r and D_2 have the same orientation in that path and this may or may not

be acceptable. Now if r is assigned to be of opposite direction to D_1 in a path k and of opposite direction to D_2 in the same path k, we find a contradiction in the orientation since D_1 and D_2 are by hypothesis of opposite direction: the only way to satisfy such a requirement is to duplicate the diode and place them back to back in that path. We then have two methods of duplication; diode duplication or contact duplication.

A constraint due to the specification is imposed on the detectors D_1 and D_2: these must form a closed path. There exists vectors in T_{12} and T_{21} which are common to each other: each common path is represented in H by two rows, one affixed with D_1 and the other affixed with D_2. This ensures that D_1 and D_2 form a closed path. On splitting diodes or contacts an arrangement must be made such that these rows remains identical or are altered in such a way that they remain identical. In resume then, we can run up against 4 possible situations in checking H for acceptability.

1) Path acceptable (covers a row of H): no diode present in that path.

2) Path acceptable only if diode is oriented in same direction as its detector in that path set.

3) Path acceptable only if diode orientation is opposite that of its detector: this implies that this path conducts for the second detector: this may or may not be acceptable. If not acceptable, diode is duplicated and must be in a back to back connection in that loop.

4) Diode duplicated to make path acceptable as in (2) or (3).

With these constraints, it is possible to check H, from the modulo 2 point of view: the final orientation is done on the unoriented graph, according to the restrictions brought about in checking H. Even if we were using an oriented matrix failure to realize the function, may be due to conflicting orientation. The artifice of using two detectors permits us to check H in the modulo 2 field; no need for maximum loop techniques are needed since there is no requirement to orient H. In checking H, only those row sums which have duplication requirements are used in making H acceptable: once a non-oriented graph has been found, all orientation requirements are imposed on the non-oriented graph.

SECTION V (c)

Example 1: $T_{12} = x'y'z' + xyz'$, $T_{21} = x'y'z' + x'yz$

$F = x'y'z'(D_1+D_2) + xyz'rD_1 + x'yzrD_2$

where r represents the diode in the unoriented state.

H =		x	x'	y	y'	z	z'	r	D_1	D_2
	1	.	1	.	1	.	1	.	1	.
	2	.	1	.	1	.	1	.	.	1
	3	.	1	1	.	1	.	1	.	1
	4	1	.	1	.	.	1	1	1	.
	123	.	1	1	⊡	1	⊡	1	1	.
	124	1	⊡	1	⊡	.	1	1	.	1
	134	1	⊡	⊡	1	1	⊡	⊡	.	1
	234	1	⊡	⊡	1	1	⊡	⊡	1	.

Row Sums	Splitting
123	y', z', r and D_1 opposite
124	x', y', r and D_2 opposite
134	x', y, z', split r and back to back connection
234	x', y, z' split r and back to back connection

On examining this table, it is seen that splitting r, or z', or y can possibly lead to a solution: this will satisfy row sums 134, 234; the first two (123, 124) can be satisfied by proper orientation: thus a minimal solution would be tired, splitting only 1 element.

All these alternatives must be tried if we are searching for a minimal solution: each duplication leads to no solution; contradictions arise at an early stage of the realization procedure. The next approach would be to duplicate a contact; i. e. solution by splitting r and x' or r and y or r and z'. Only the solution for r and x' is given.

Split r and x'

:constraint on x' splitting, such that rows 1 and 2 of H are not altered or, if altered they remain identical except for D_1 and D_2:

H =		x	x'_1	x'_2	y	y'	z	z'	r_1	r_2	D	D_2
	1	.	1	.	.	1	.	1	.	.	1	.
	2	.	1	.	.	1	.	1	.	.	.	1
	3	.	.	1	1	.	1	.	.	1	.	1
	4	1	.	.	1	.	.	1	1	.	1	.

r_2 and D_1 opposite 1 2 3

r_1 and D_2 opposite in 1 2 4

$$B_f = \begin{array}{c} \\ 12 \\ 2 \\ 3 \\ 14 \end{array} \begin{array}{c} \begin{array}{cccc} D_1 & z' & z & x \end{array} \\ \left[\begin{array}{cccc} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{array}\right. \end{array} \begin{array}{c} \begin{array}{cccccc} x'_1 & x'_2 & y & y' & r_1 & r_2 \end{array} \\ \left|\begin{array}{cccccc} . & . & . & . & . & . \\ 1 & . & . & 1 & . & . \\ . & 1 & 1 & . & . & 1 \\ 1 & . & 1 & 1 & 1 & . \end{array}\right. \end{array} \begin{array}{c} D_2 \\ \left.\begin{array}{c} 1 \\ 1 \\ 1 \\ . \end{array}\right] \end{array}$$

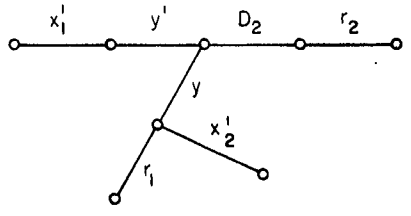
$K_2 =$ $\begin{array}{ccc} x' & y' & D_2 \end{array}$ ordered as is, $K_3 =$ $\begin{array}{ccc} x'_2 & y & r_2 \end{array}$ D_2 ordered as is

$$\begin{bmatrix} . & . & 1 \\ 1 & 1 & 1 \\ . & . & 1 \\ 1 & 1 & . \end{bmatrix} \qquad \begin{bmatrix} . & . & . & 1 \\ . & . & . & 1 \\ 1 & 1 & 1 & 1 \\ . & 1 & . & \end{bmatrix}$$

$K_4 =$ $\begin{array}{cccc} x'_1 & y & y' & r_1 \end{array}$ reordered as $x'_1 y' y r_1$

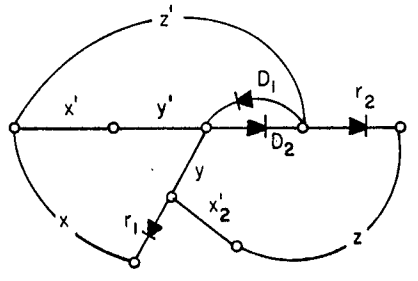
$$\begin{bmatrix} . & . & . & . \\ 1 & . & 1 & . \\ . & 1 & . & . \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

we then get



as a tree

and

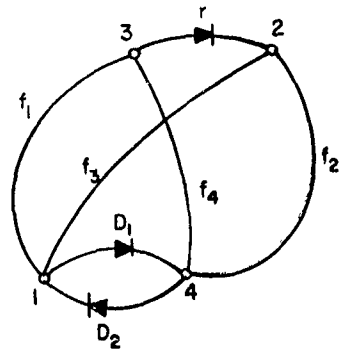


as the oriented graph.

(the orintation procedure according to constraints on matrix and F)

It would seem that this solution is minimal: however; by a simple topological process another solution will be presented.

Consider



Where f_i is path between nodes J and K

Call loop 1 = $f_1 r f_2 D_2$

Call loop 2 = $f_4 r f_3 D_1$

if $f_1 f_4 = 0$, $f_3 f_2 = 0$ disjunctive then loops 1 and 2 are the only possible conducting loops.

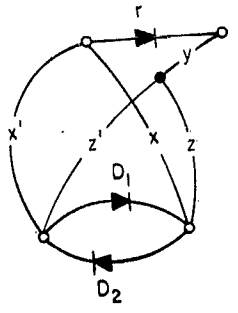
: given $f_1 = x'$, and $f_4 = x$

and $f_2 = y z$, and $f_3 = z' y$

We have then $f_1 r f_2 D_2 = x' y z r D_2$

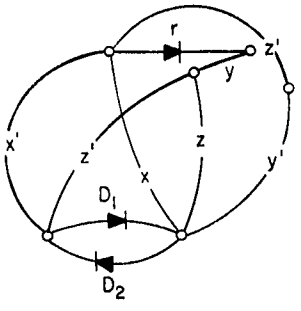
and $f_4 r f_3 D_1 = x y z' r D_1$

It remains to add the bilateral loop.



as the non-bilateral portion of the graph

: adding the bilateral term we get, on duplicating z'



as the complete graph.

This solution presents only 1 contact duplication: duplicating x' instead of z' in the bilateral path leads to a similar solution. The conclusion to this example being that even after only 1 contacts is split, methods of extending the vector space must be tried, and this is never an obvious step[15].

Ex: 2 Realize

$$\left\{ T_{12} = abc + ac + ae + cd + de, \qquad T_{21} = abc + abe + bcd + de \right\}$$

$$F = abc(D_1 + D_2) + de(D_1 + D_2) + (abc + bcd)D_2 + (ae + ac + cd)rD_1$$

On examination of this specification we find no primed variable. If the loop-set matrix contains unacceptable paths a method available to make paths acceptable, is to orient the diode and the detector in opposite directions or duplicate diodes. In the specification of multi-terminal metworks, mention should be made of the importance of linearly dependent paths, once a duplication of contact is made. This example is an illustration of this fact with diodes.

$H =$

	a	b	c	d	e	r	D_1	D_2
1	1	1	1	.	.	.	1	.
2	1	1	1	.	.	.	.	1
3	.	.	.	1	1	.	.	1
4	1	.	.	.	1	1	1	.
5	1	.	1	.	.	1	1	.
6	.	.	1	1	.	1	1	.
7	1	1	.	.	1	.	.	1
8	.	1	1	1	.	.	.	1
9	.	.	.	1	1	.	1	1

Rows 7, 8, and 9 are linearly dependent and in checking H are deleted. They are brought in question only if there occurs a duplication in any of the rows they are linearly dependent on. Linearly dependent rows 7 = 245, 8 = 256, 9 = 123 (modulo 2). Linear combinations of rows modulo 2 are taken, 3 rows, 5 rows at a time and unacceptable paths are tabulated.

Row sums	Requirements
124	r and D_2 in opposite direction
125	"
126	"
156	r and D_1 in opposite direction
345	split r: r and D_2 in opposite direction
356	" "
356	" "
12345	" "
12346	" "
12356	" "

Requirements: split r and r and D_2 in opposite direction is the case of a path made acceptable by insertion of a diode to make this path cover a row of H.

Duplication is performed: then orientation requirements are checked.

Splitting is of the type

$$\begin{array}{c} \\ 4 \\ 5 \\ 6 \end{array}\begin{array}{c} \begin{array}{cc} r_1 & r_2 \end{array} \\ \begin{bmatrix} 1 & 1 \\ 1 & . \\ . & 1 \end{bmatrix} \end{array} \qquad \begin{array}{c} \begin{array}{cc} r_1 & r_2 \end{array} \\ \begin{bmatrix} 1 & . \\ 1 & 1 \\ . & 1 \end{bmatrix} \end{array} \qquad \begin{array}{c} \begin{array}{cc} r_1 & r_2 \end{array} \\ \begin{bmatrix} 1 & . \\ . & 1 \\ 1 & 1 \end{bmatrix} \end{array}$$

(a) (b) (c)

Each of these alternatives can be tried: a non-oriented graph may result, and the orientation requirements may be satisfied. A more logical way of proceeding, is to examine the linearly dependen paths.

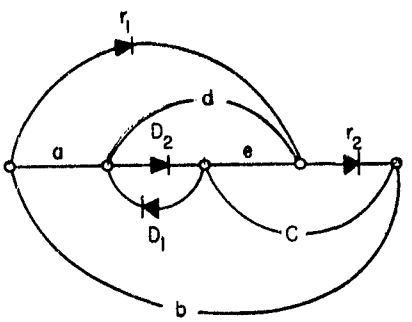
case (a) Path 8 = 256 = $bcdr_1r_2D_1$: this implies, r_1, r_2, D_2 in the same direction
A subset of this path: 1345 = cdr_2D_1

This cannot be satisfied by any orientation of the r's.

case (c) Path 7 = 245 = $aber_1r_2D_2$: this implies, r_1, r_2, D_2 in the same direction.
A subset of this path: 12356 = aer_1D_1
This cannot be satisfied by any orientation of the r's.

case (b) No linearly dependent path has both diodes in it. This case is used in H. The matrix is diagonalized : the orientation proceeds as according to the requirements.

$$B_f = \begin{array}{c} \begin{array}{ccccccccc} c & D_1 & d & r_1 & b & a & D_2 & e & r_2 \end{array} \\ \begin{bmatrix} 1 & & & & & . & . & 1 & 1 \\ & 1 & & & & . & 1 & . & . \\ & & 1 & & & . & 1 & 1 & . \\ & & & 1 & & 1 & 1 & 1 & . \\ & & & & 1 & 1 & 1 & 1 & . \end{bmatrix} \end{array} \begin{array}{r} 56 \\ 12 \\ 4 \\ 467 \\ 256 \end{array}$$



If an argument by linearly dependent paths is not possible, then the procedure would be to use each alternative; if no orientation is consistent with the specification, another alternative must be used. The realization procedure is then two-fold: finding a non-oriented graph and finding an orientation to the graph; the realization may fail due to any of these steps.

The whole argument in this presentation is based on methods that avoid the oriented matrix and generalize the tree configuration possible: if the non-oriented matrix is unrealizable, then of course so is the oriented one; if the oriented matrix is not regular, then the non-oriented matrix may still be realizable[14]. Since only the diodes and detectors need be oriented, the artifice of using two detectors is sufficient to yield enough information for realizability.

SECTION VI Conclusions

A method of finding conditions of realizability of fundamental circuit matrices has been developed and used extensively throughout this work. The process is mechanical up to the point of combining various linear trees that it yields. Failure due to unrealizability of a sub-matrix is immediately recognized: failure due to impossibility of constructing a graph satisfying the order of each linear tree is still a semi-mechanical visual process. It is felt that even this step could be mechanized completely.

The examples on constrained variables matrices were brought up because of their recurrence as soon as problems of moderate size are undertaken. The multi-terminal examples were introduced and developed up to working point because of the inherent possibility of getting minimal solutions: as was seen the method is so geared as to search gradually for a solution; if a quick solution is wanted, more than the minimum number of duplications are performed.

Simple one port diode-contact networks were analyzed and synthesized as an outgrowth of the methods developed for multiple-output synthesis. The process was essentially that of orienting the graph after realization modulo 2, and using a set of new constraints on the matrix.

The graph theory approach is not at the moment the most flexible method of realization because of its character of minimality. The procedures developed seem to point the way to total mechanization in order to yield quickly a minimal solution, since the process is repetitive in nature, and involves for each repetition, examination, testing and realizing matrices as graphs of specified functions.

References

1. Gould, R., "The Application of Graph Theory to the Synthesis of Content Networks: dissertation", Harvard University, 1957.
2. Seshu and Reed, "Linear Graphs and Electrical Networks", Addison-Wesley, 1961.
3. Smith, Healy and Mow, "Synthesis of Two-Terminal Contact Diode Networks", P. I. B. report, 1962.
4. Gould, R., "Graph and Vector Spaces", J. Math and Physics # 38, 1958.
5. Mayeda, W., "Synthesis of Switching Functions by Linear Graph Theory". I. B. M. Journal of Research and Development, V-4, p. 321-28, 1960.
6. Ash and Kim, "Realizability of a Circuit Matrix". I. R. E., P. G. C. T., June 1959.
7. Mayeda, W., "Necessary and Sufficient Conditions for Realizability of Cut-Set Matrices", I. R. E., P. G. C. T., March 1960.
8. Lofgren, L., "Inredundant and Redundant Boolean Branch Networks". I. R. E., P. G. C. T. -6, May 1959.
9. Okada and Young, "Ambit Realization of Cut-Set Matrices into Graphs", Research Report, Microwave Research Institute of Brooklyn Polytechnic Institute. no., R-790-59, 1959.
10. Halkias and Kim, "Realization of Fundamental Circuit and Cup-Set Matrices", I. R. E. Internation Convention, 1962.
11. Guillemin, E. A., "How to Grow Your Own Trees From Given Cut-Set or Tie Set Matrices", I. R. E., P. G. C. T. Special Supplement. May 1959.
12. Caldwell, S. H., "Switching Circutis and Logical Desing", Wiley. 1958.
13. Vandling, "Simplication of Multiple Output Switching Netowrks Composed of Uni lateral Devices", I. R. E., P. G. E. C., Dec. 1960.
14. Seshu, "Realizability of Directed Graphs", I. R. E., P. G. C. T., March 1962.
15. Okada, Moiuraki, and Young, Research Report R-790, P. I. B. 718, Polytechnic Institute of Brooklyn.

TECHNICAL AND FINAL REPORT DISTRIBUTION LIST

PHYSICAL SCIENCES DIRECTORATE

Commander 3
AF Office of Scientific Research
ATTN: SRY
Washington 25, D. C.

Commander 2
AF Office of Scientific Research
ATTN; SRGL
Washington 25, D. C.

Commander 4
Wright Air Development Division
ATTN: WWAD
Wright-Patterson Air Force Base
Ohio

Commander 1
AF Cambridge Research Laboratories
ATTN: CRRELA
L. G. Hanscom Field
Bedford, Massachusetts

Commander 2
European Office
Office of Aerospace Research
The Shell Building
Brussels, Belgium

P. O. Box AA 1
Wright-Patterson Air Force Base
Ohio

Aeronautical Research Laboratories 1
ATTN: Technical Library
Building 450
Wright-Patterson Air Force Base
Ohio

Armed Services Technical Info. Agency
ATTN: TIPCR 10
Arlington Hall Station
Arlington 12, Virginia

Director of Research and Development
Headquarters, USAF 1
ATTN: AFDRD
Washington 25, D. C.

Office of Naval Research 1
Department of the Navy
ATTN: Code 420
Washington 25, D. C.

Director, Naval Research Laboratory
ATTN: Technical Info. Officer 1
Washington 25, D. C.

Chief of Research and Development 1
ATTN: Scientific Information Branch
Department of the Army
Washington 25, D. C.

Chief, Physics Branch 1
Division of Research
U. S. Atomic Energy Commission
Washington 25, D. C.

U. S. Atomic Energy Commission 1
Technical Information Extension
P. O. Box 62
Oak Ridge, Tennessee

National Bureau of Standards Library 1
Room 203, Northwest Building
Washington 25, D. C.

Physics Program 1
National Science Foundation
Washington 25, D. C.

Director, Office of Ordnance Research
Box CM, Duke Station 1
Durham, North Carolina

Director, Department of Commerce 1
Office of Technical Services
Washington 25, D. C.

AEDC (AEOIM) 1
Arnold Air Force Station
Tullahoma, Tennessee

Commander 1
AF Flight Test Center
ATTN: FTOTL
Edwards Air Force Base, California

Commander 1
AF Special Weapons Center
ATTN: SWOI
Kirtland Air Force Base, New Mexico

Commander 1
AF Missile Development Center
ATTN: HDOI
Holloman Air Force Base
New Mexico

Commander 1
Army Rocket and Guided Missile Agency
ATTN: ORDXR-OTL
Redstone Arsenal, Alabama

Commandant Air Force Institute of Technology AU Library MCLI-LIB, Bldg. 125, Area B Wright-Patterson Air Force Base Ohio	1
Commander Air Force Systems Command ATTN: RDRS Andrews Air Force Base Washington 25, D. C.	6
Commanding General U. S. Army Signal Corps Research and Development Laboratory ATTN: SIGFM/EL-RPO Ft. Monmouth, New Jersey	1
National Aeronautics and Space Administration ATTN: Library 1520 H Street, N. W. Washington 25, D. C.	6
Advanced Research Projects Agency Washington 25, D. C.	1
Rand Corporation 1700 Main Street Santa Monica, California	1
Chairman (Unclass. Reports) Canadian Joint Staff For DRB/DSIS 2450 Massachusetts Ave., N. W. Washington 25, D. C.	1